



TITLE:

制約付き非線形最適化手法の一提案 : α 制約遺伝的アルゴリズム (最適化の数理とアルゴリズム)

AUTHOR(S):

阪井, 節子; 高浜, 徹行

CITATION:

阪井, 節子 ...[et al]. 制約付き非線形最適化手法の一提案 : α 制約遺伝的アルゴリズム (最適化の数理とアルゴリズム). 数理解析研究所講究録 2002, 1297: 28-37

ISSUE DATE:

2002-12

URL:

<http://hdl.handle.net/2433/42650>

RIGHT:

制約付き非線形最適化手法の一提案

— α 制約遺伝的アルゴリズム—

広島修道大学商学部 阪井 節子 (Setsuko Sakai)

Faculty of Commercial Science, Hiroshima Shudo University

広島市立大学情報科学部 高濱 徹行 (Tetsuyuki Takahama)

Faculty of Information Sciences, Hiroshima City University

1 まえがき

与えられた制約のもとで目的関数を最小にするような解を求める制約付き最適化問題、特に非線形最適化問題は、実問題に頻繁に出現する重要な最適化問題である。制約付き最適化問題の解法としては、線形計画問題の場合はシンプレックス法がよく知られており、目的関数が非線形で微分可能な場合には、目的関数の2次近似式と制約式の線形近似式により得られる2次計画問題を逐次解く逐次2次計画法、目的関数の最急降下方向やその修正方向を有効制約式の作るアフィン集合上に射影した方向に探索する射影法、線形計画法におけるシンプレックス法を非線形計画問題に拡張した一般縮小勾配法などの最適化手法が知られている[1, 2]。しかし、目的関数が微分可能でない場合や制約条件が非凸の場合には、これらの方法を利用することは困難である。

より汎用的な方法としては、制約付き最適化問題を制約のない最適化問題に変換する変換法と目的関数値のみを用いる直接探索法を組み合わせることが有効であり、多くの応用例が存在する。変換法[3]には、ペナルティ法、乗数法などがあるが、制約条件の境界付近に解が存在する場合が多いこと、パラメータが少なく単純で使用しやすいことなどからペナルティ法を採用することが多いが、ペナルティ係数の値が大きくなるにつれて、変換された問題の最適化が困難になるという問題、制約条件を完全に満足する実行可能解を探索することが困難であるという問題がある[4]。

一方、直接探索法の一つとして、近年多点探索を行う遺伝的アルゴリズム (Genetic Algorithm, GA)[5, 6]が広く使用されるようになってきている。GAは、生物の進化過程を模倣した最適化アルゴリズムであり、強力かつ柔軟な最適化手法として注目されている。GAを利用した制約付き最適化に関する研究も盛んに行われるようになってきており[7]、既存の方法と比較しても遜色のない結果が得られるようになってきている[8]。特にMichalewiczらは、遺伝子を実数値とする実数値GA(real-coded GA)[6, 9, 10]を採用し、すべての制約式を満足する個体の集団である参照点(reference point)集団と線形制約式のみを満足する探索(search point)集団の2つの集団を用いることにより制約を扱う GENOCOPIII(GEnetic algorithm for Numerical Optimization of COnstrained Problems)を提案し、一般に非線形制約をも取り扱えることを示した[7, 8, 11, 12, 13]。

本研究では、 α 制約法[14, 15]を遺伝的アルゴリズムと組み合わせた α 制約遺伝的アルゴリズム(α GA)[16]を提案する。 α 制約法は、制約の満足度を表現する制約満足度を導入し、通常の大小関係の代わりに制約満足度を優先した大小関係である α レベル比較を定義し、通常と比較の代わりに α レベル比較を用いて探索することにより、制約付きの問題を制約のない問題に変換する方法である。 α 制約法の特徴は、目的関数を変換するのではなく、最適化アルゴリズムにおける比較を置換することにより、アルゴリズムを変換するという点である。文献[14]では、 α 制約法をPowell法[17]と組み合わせ、文献[15]では、Nelder&MeadのSimplex法[18]と組み合わせることにより、目的関数が微分不可能な制約付き最適化問題が効率よく解けることを示した。 α 制約法がGAにも適用可能であることを示すことも本論文の目的である。

α 制約法を適用した α GAでは、制約を満足しない個体は制約を満足するように、制約を満足した個体は目的関数値を最適化するように自然に進化する。本論文では、線形計画問題、非線形計画問題、非凸非線形制約など様々な種類のテスト問題について、GAによる制約付き最適化手法の中で有効性がよく知られている GENOCOPIIIを発展させた GENOCOP5.0[19, 20]と比較することにより、 α GAの有効性を示す。

用して解く方法の中で有効性がよく知られている GENOCOP5.0 と比較することにより, α GA がこの方法よりもさらに優れた方法であることを示した。

α GA は, 制約を満足しない場合には制約満足度が最大化され, 自然に制約を満足するようになるため, GENOCOPIII や GENOCOP5.0 とは異なり, 制約条件を保存しない遺伝的操作を導入することが可能である。

今後は, 実数値遺伝的アルゴリズムの研究で提案されている様々な遺伝的操作を有効に取り込むとともに, 様々な実問題に対して α GA を適用していく予定である。

謝辞 本研究の一部は, 日本学術振興会科学研究費補助金基盤研究 (C)(2)(課題番号 14580498) による補助のもとで行われた。

参考文献

- [1] 今野浩, 山下浩: 非線形計画法, 日科技連出版社 (1978).
- [2] 坂和正敏: 非線形システムの最適化, 森北出版 (1986).
- [3] Fiacco, A. and McCormick, G.: *Nonlinear programming : Sequential unconstrained minimization techniques*, Society for Industrial and Applied mathematics (1990).
- [4] 坂和正敏, 矢内克裕: 非凸非線形計画問題に対する浮動小数点型遺伝的アルゴリズム: 改良型 GENOCOP I, 電子情報通信学会論文誌, Vol. J81-A, No. 1, pp. 90–97 (1998).
- [5] Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989).
- [6] 坂和正敏, 田中雅博: 遺伝的アルゴリズム, 朝倉書店 (1995).
- [7] Michalewicz, Z.: A Survey of Constraint Handling Techniques in Evolutionary Computation Methods, *Proc. of the 6th International Conference on Genetic Algorithms*, Cambridge, MA, MIT Press, pp. 135–155 (1995).
- [8] Michalewicz, Z.: *Genetic algorithm + data structures = evolution programs 3rd ed.*, Springer-Verlag, Berlin (1996).
- [9] Tsutsui, S., Yamamura, M. and Higuchi, T.: Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms, *Proc. Genetic and Evolutionary Computation Conference (GECCO'99)*, pp. 657–664 (1999).
- [10] 小野功, 山村雅幸, 喜多一: 実数値 GA とその応用, 人工知能学会誌, Vol. 15, No. 2, pp. 259–266 (2000).
- [11] Michalewicz, Z.: Genetic Algorithms, Numerical Optimization and Constraints, *Proc. of the 6th International Conference on Genetic Algorithms*, Pittsburgh, pp. 151–158 (1995).
- [12] Michalewicz, Z., and Nazhiyath, G.: Genocop III: A Co-evolutionary Algorithm for Numerical Optimization Problems with Nonlinear Constraints, *Proc. of the 2nd IEEE International Conference on Evolutionary Computation*, Vol. 2, Perth, pp. 647–651 (1995).
- [13] Michalewicz, Z. and Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization Problems, *Evolutionary Computation*, Vol. 4, No. 1, pp. 1–32 (1996).

以下, 2. で本論文の対象とする制約付き最適化問題を定義し, 3. で α 制約法を説明する. 4. で α GA を定義し, 5. で α GA の性能を示す. 6. はまとめである.

2 制約付き最適化問題

本論文では, 次のような不等式制約, 等式制約, 上下限制約を持つ最適化問題 (P) を考える. 目的関数および制約条件がともに線形の場合が線形計画問題, その他の場合が非線形計画問題である.

$$\begin{aligned}
 \text{(P)} \quad & \text{minimize } f(\mathbf{x}) \\
 & \text{subject to } g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, q \\
 & \quad \quad \quad h_k(\mathbf{x}) = 0, \quad k = 1, \dots, m - q \\
 & \quad \quad \quad l_i \leq x_i \leq u_i, \quad i = 1, \dots, n
 \end{aligned} \tag{1}$$

ここで, $\mathbf{x} = (x_1, \dots, x_n)$ は n 次元決定変数ベクトル, $f(\mathbf{x})$ は 目的関数, $g_j(\mathbf{x}) \leq 0$ は q 個の不等式制約, $h_k(\mathbf{x}) = 0$ は $m - q$ 個の等式制約であり, f, g_j, h_k は線形あるいは非線形の実数値関数である. l_i, u_i はそれぞれ, 決定変数 x_i の下限値, 上限値である. さらに, 以下では実行可能領域を F , 上下限制約のみを満足する領域を S と呼ぶことにする.

3 α 制約法

α 制約法について簡単に説明する.

3.1 制約付き最適化問題と制約満足度

α 制約法では, 制約をどの程度満足しているかを表現するために, 制約満足度 $\mu(\mathbf{x})$ を導入する. 制約満足度 $\mu(\mathbf{x})$ は, 以下を満足する関数である.

$$\begin{cases} \mu(\mathbf{x}) = 1, & \text{if } g_j(\mathbf{x}) \leq 0, \quad h_k(\mathbf{x}) = 0, \quad \text{for all } j, k \\ 0 \leq \mu(\mathbf{x}) < 1, & \text{otherwise} \end{cases} \tag{2}$$

このような制約満足度を定義する 1 つの方法として, 各制約の満足度を定義し, それらを合成する方法がある. 例えば, 問題 (P) における各制約条件は, 機械的に以下のような g_j, h_k に関する区分的線形の制約満足度関数に変換できる. ただし, $b_j, b_k (> 0)$ は適当な定数である.

$$\mu_{g_j}(\mathbf{x}) = \begin{cases} 1, & \text{if } g_j(\mathbf{x}) \leq 0, \\ 1 - \frac{g_j(\mathbf{x})}{b_j}, & \text{if } 0 \leq g_j(\mathbf{x}) \leq b_j, \\ 0, & \text{otherwise,} \end{cases} \quad \mu_{h_k}(\mathbf{x}) = \begin{cases} 1 - \frac{|h_k(\mathbf{x})|}{b_k}, & \text{if } |h_k(\mathbf{x})| \leq b_k, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

この各制約満足度から制約全体の満足度 $\mu(\mathbf{x})$ を求めるために, 各制約満足度を結合する. 結合演算としては, 以下のような min 演算が考えられる.

$$\text{min 演算} \quad \mu(\mathbf{x}) = \min_{j,k} \{ \mu_{g_j}(\mathbf{x}), \mu_{h_k}(\mathbf{x}) \} \tag{4}$$

3.2 α レベル比較

関数値と制約満足度の組 (f, μ) の集合上において, 制約満足度が α 以上の場合は目的関数値の大小関係を優先し, それ以外の場合は制約満足度の大小関係を優先する比較である α レベル比較を定義する.

点 x_1, x_2 における関数値を f_1, f_2 , 制約満足度を μ_1, μ_2 とすると, 通常的大小関係である \leq に対応する関数値と制約満足度の組 (f_i, μ_i) 間的大小関係である α レベル比較 \leq_α ($0 \leq \alpha \leq 1$) は以下ようになる.

$$(f_1, \mu_1) \leq_\alpha (f_2, \mu_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \mu_1, \mu_2 \geq \alpha \\ f_1 \leq f_2, & \text{if } \mu_1 = \mu_2 \\ \mu_1 > \mu_2, & \text{otherwise} \end{cases} \quad (5)$$

なお, \leq_0 は関数値のみの比較と一致し, \leq_1 は制約満足度を優先する辞書式比較と一致する.

3.3 α 制約法の性質

α 制約法は, 制約付き最適化問題を直接探索法で解く際に, 通常と比較の代わりに α レベル比較を用いる方法である. 通常的大小比較を α レベル比較に置き換えた最適化問題 (P_{\leq_α}) , すなわち, α 制約法による最適化問題は以下のように定義できる. 但し, $\text{minimize}_{\leq_\alpha}$ は \leq_α の意味での最小化である.

$$(P_{\leq_\alpha}) \quad \text{minimize}_{\leq_\alpha} \quad f(x) \quad (6)$$

ここで, 問題 (P) の制約条件を $\mu(x) \geq \alpha$ に緩和した問題 (P^α) を以下のように定義する. なお, (P^1) は問題 (P) と等価である.

$$\begin{aligned} (P^\alpha) \quad & \text{minimize} \quad f(x) \\ & \text{subject to} \quad \mu(x) \geq \alpha \end{aligned} \quad (7)$$

問題 (P^α) と問題 (P_{\leq_α}) , および問題 (P) に関して以下の定理が成り立つ [14].

定理 1 問題 (P^1) に最適解が存在するならば, 問題 (P_{\leq_α}) の最適解は問題 (P^α) の最適解である.

定理 2 問題 (P) に最適解が存在するならば, 問題 (P_{\leq_1}) の最適解は, 問題 (P) の最適解である.

定理 3 $\{\alpha_n\}$ を強い意味で単調増加し 1 に収束する点列とする. $f(x), \mu(x)$ を連続関数とし, 問題 (P^1) の最適解 x^* の存在と任意の α_n に対する問題 $(P_{\leq_{\alpha_n}})$ の最適解 \hat{x}_n の存在を仮定する. このとき, 点列 $\{\hat{x}_n\}$ の任意の集積点は問題 (P^1) の最適解である.

定理 1, 2 は, α レベル比較を行うことにより, 制約付き問題がそれと等価な制約のない問題に変換されることを示している. したがって, 既存の制約のない最適化手法に α レベル比較を導入することにより, 制約付きの問題を解くことが可能となる. 定理 3 は, ペナルティ法においてペナルティ係数を ∞ まで増加させて行くのと同様に, α を 1 まで増加させながら最適化を行っても, 最適解が得られることを示している.

4 α 制約遺伝的アルゴリズム α GA

α 制約法は, 目的関数値の大小関係のみに基づく最適化アルゴリズムと組み合わせることができる. 本研究では, 適合度の大小関係のみに依存する戦略であるランキング戦略を採用することにより, 遺伝的アルゴリズムに α 制約法を適用する.

4.1 アルゴリズム

α GA のアルゴリズムを以下に示す.

1. 初期集団の生成: 初期個体集団をランダムに生成する. すなわち, 各遺伝子 x_i を区間 $[l_i, u_i]$ の一様乱数で生成する.

2. 終了判定：本論文では最大世代数 T に達したとき、実行を終了する。
3. α レベルの決定：通常は $\alpha = 1$ として探索すればよいが、等式制約のように実行可能領域が非常に狭い場合には、実行可能領域が狭すぎて有効な最適化が行えない。このような場合には α レベルを制御し、初期には制約条件を緩和して目的関数の最適化を優先し、次第に制約条件を厳しくしてゆく必要がある (4.2 参照)。
4. 選択：個体の選択方法として、適合度の大小比較のみに依存する方法であり、頑健な方法として知られているランキング選択を採用する (4.3 参照)。
5. 交叉：交叉確率 P_c で親の交叉を行い、子を生成する。なお、交叉しない場合には親がそのまま子になったとみなす。本論文では、実数値 GA 用の交叉を採用する (4.4 参照)。
6. 突然変異：本論文では、正規分布にしたがった突然変異であるガウス突然変異 (Gaussian mutation) と制約付き最適化に有効であると考えられる境界突然変異 (boundary mutation) を突然変異として採用する (4.5 参照)。
7. 世代交代：現在の集団を子に置き換え、2. へ戻る。

α GA のアルゴリズムを以下に C 言語風に記述する。

```

 $\alpha$ GA()
{
    t=0;
    個体集団  $P(t)$  を生成;
    while( $t \leq T$ ) {
         $\alpha = \alpha(t)$ ;
         $\alpha$  レベル比較により  $P(t)$  中の個体をランク付け;
         $P' = P(t)$  からランキング選択;
        for(each  $m$  個体組  $x^1, \dots, x^m$  in  $P'$ ) {
            確率  $P_c$  で  $x^1, \dots, x^m$  を交叉;
            for(each 遺伝子  $x_i$  in  $x^1, \dots, x^m$ ) {
                確率  $P_b$  で  $x_i$  を境界突然変異;
                確率  $P_G$  で  $x_i$  をガウス突然変異;
            }
        }
        t = t + 1;
         $P(t) = P'$ ;
    }
}

```

ただし、 $\alpha(t)$ は世代 t における α レベル、 P_c は交叉率、 P_b は境界突然変異率、 P_G はガウス突然変異率である。

4.2 α レベルの制御

本論文では、等式制約を含む最適化問題に対して、次のような制御方法を採用する。 α レベルを満足する個体が常にある程度含まれるように、最初は速く、1 に近づくほどゆっくりと増加し、1 になった後もあ

る程度の最適化を行うことを目指す。そこで、初期値 $\alpha(0)$ を初期集団の制約満足度の平均値と最大値の間とし、最大世代数 T の半分以上は常に 1 となる以下のような世代 t の 2 次式に基づく制御を提案する。

$$\alpha(0) = \frac{1}{2}(\max_{\mathbf{x}_k} \mu(\mathbf{x}_k) + \frac{1}{N} \sum_{k=1}^N \mu(\mathbf{x}_k)), \quad (8)$$

$$\alpha(t) = \begin{cases} -(1 - \alpha(0))(1 - \frac{2t}{T})^2 + 1, & 0 < t < \frac{T}{2}, \\ 1, & t \geq \frac{T}{2}. \end{cases}$$

ただし、 N は個体集団サイズ、 $\mathbf{x}_k (k = 1, 2, \dots, N)$ は各個体である。

4.3 選択

ランキング選択にはいくつかの種類があるが、本論文では線形ランキング選択を用いる [21, 22]。すなわち、個体集団中において α レベル比較によりランク付けを行い、ランクに応じた確率で選択確率を決定する。個体 \mathbf{x}_k のランクが $r_k (r_k = 1, 2, \dots, N)$ となった場合、個体 \mathbf{x}_k の選択確率 p_k は以下のように決定される。

$$p_k = \frac{1}{N} \left(\eta^+ - (\eta^+ - \eta^-) \frac{r_k - 1}{N - 1} \right) \quad (9)$$

ただし、 $\eta^- = 2 - \eta^+$ 、 η^+ は、最大期待値 (maximum expected value) であり、最良の個体が平均的個体より何倍選択されるかを指定する区間 $[1.0, 2.0]$ の数である。

4.4 交叉

Michalewicz は実行可能領域が凸の場合には実行可能性が保存される交叉である、単純交叉 (simple crossover)、算術交叉 (arithmetic crossover)、ヒューリスティック交叉 (heuristic crossover) という 3 種類の交叉を提案している。これに対して、 α GA では交叉によって制約が満足されなくなった場合には、制約満足度の最大化が優先され、自然に制約を満足する方向に最適化される。したがって、実行可能性の保存をそれほど重要視する必要はない。そこで、本論文では親個体によって張られる単体 (simplex) に相似な単体内に一様分布にしたがって、子の個体を生成することによって、子の多様性を維持するシンプレックス交叉 (SPX; simplex crossover) [9, 10] を用いる。

m 個の親個体 $\mathbf{x}^1, \dots, \mathbf{x}^m$ が交叉するとき、交叉後の子個体 $\mathbf{x}^{1'}, \dots, \mathbf{x}^{m'}$ を以下のように定める。

- (1) 親の重心を $\mathbf{g} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}^i$ とする。
- (2) $\mathbf{y}^i, i = 1, \dots, m$ を次式で求める: $\mathbf{y}^i = \mathbf{g} + \beta_c (\mathbf{x}^i - \mathbf{g})$ で求める。ここで、ただし、 β_c は子個体が親個体の範囲からどの程度広がるかを定める正の定数である。
- (3) $\mathbf{y}^i, i = 1, \dots, m$ で張られた単体 $K\{\mathbf{y}^1, \dots, \mathbf{y}^m\}$ 内に一様に m 個の子個体 $\mathbf{x}^{1'}, \dots, \mathbf{x}^{m'}$ を生成する。

4.5 突然変異

Michalewicz は実行可能性を保存する突然変異として、境界突然変異、一様突然変異 (uniform mutation)、非一様突然変異 (non-uniform mutation)、全体非一様突然変異 (whole non-uniform mutation) という 4 種類の突然変異を提案している。本論文では、これらを参考にして、境界突然変異、ガウス突然変異の 2 種類を採用する。

以下の突然変異では、個体 \mathbf{x} のある一つの遺伝子 x_i のみを変化させ、新しい個体 $\mathbf{x}' = (\dots, x_{i-1}, x'_i, x_{i+1}, \dots)$ を得る。

1. 境界突然変異 (boundary mutation)

境界突然変異では、ある遺伝子 x_i のみを変化させたときに、新しい個体の実行可能領域 F に存在するときの端点を以下のように $l(x_i)$, $r(x_i)$ とし、 x_i をどちらかに等確率で置換する。

$$\begin{aligned} l(x_i) &= \min_{(\dots, x_{i-1}, x'_i, x_{i+1}, \dots) \in F} x'_i \\ r(x_i) &= \max_{(\dots, x_{i-1}, x'_i, x_{i+1}, \dots) \in F} x'_i \end{aligned} \quad (10)$$

α GA では、実行可能領域中に存在する個体とそうでない個体が混在しており、実行可能でない個体では、 $l(x_i)$ や $r(x_i)$ が存在しない場合もある。このため、個体の実行可能ときには上記の突然変異を行い、実行可能でないときには実行可能領域に入れるために制約満足度 μ を最大化する。すなわち、境界突然変異率 P_b で以下のように x_i を x'_i に置換することになる。

$$x'_i = \begin{cases} l(x_i) \text{ or } r(x_i) & x \in F \\ \arg \max_{l_i \leq x'_i \leq u_i} \mu(\dots, x'_i, \dots) & x \notin F \end{cases} \quad (11)$$

なお、 $l(x_i)$, $r(x_i)$ の探索と μ の最大化には、囲い込み法と2分法による直接探索を用いた。境界突然変異は、制約付き最適化問題の解が実行可能領域の境界付近に存在することが多いことから、このような解の探索に有効であると考えられる。

2. ガウス突然変異 (Gaussian mutation)

境界突然変異は比較的大域的な探索を行うため、局所的な探索を積極的に行うガウス突然変異を併用する。各変数 x_i の上下限制約の幅に比例した標準偏差を持つガウス分布に基づき、ガウス突然変異率 P_G で以下のように x_i を x'_i に置換する。

$$x'_i = x_i + \delta \quad (12)$$

ただし、 δ は $N(0, \beta_G^2(u_i - l_i)^2)$ の乱数、 β_G はガウス突然変異の分散を決めるパラメータである。

5 制約付き非線形計画問題

本論文では、Michalewicz の提案した最適化問題を対象に最適化を行い、その結果について比較検討する。

5.1 テスト問題と実験条件

表1に Michalewicz が提案した5つの問題 $G_1 \sim G_5$ [8, 11] を示す。 G_1 は線形計画問題、その他は非線形計画問題であり、 G_4 は等式制約を含んでいる。これらの問題について、 α GA を GENOCOP5.0 と比較する。GENOCOP5.0 は全ての問題について標準設定を用いた。すなわち、探索点集団と参照点集団サイズを各70、最大世代数 $T = 5000$ 、非線形ランキング選択のパラメータ $c = 0.1$ 、単純交叉、算術交叉、ヒューリスティック交叉、境界突然変異、非一様突然変異、全体非一様突然変異の確率を $4/70$ とした。

なお、 G_4 では、制約領域が非常に狭いため標準設定では参照点を求めることができず、また参照点を与えても最適化が進まなかった。そこで、Michalewicz の提案に基づき、等式制約の条件を緩和し、以下の不等式制約に置き換えた。

$$|h_j(x)| \leq \epsilon, \epsilon > 0 \quad (13)$$

さらに、参照点を求めることができなかった場合には、Simplex法を用いて等式制約関数の2乗和 $\sum_j h_j(x)^2$ を最小化することにより参照点を求める処理を追加して実験を行った。ただし、ある程度の最適化を行えるようにやや緩い制約 $\epsilon = 10^{-3}$ に設定した。

表 1: Michalewicz のテスト問題

$$G_1(x) = \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i,$$

subject to

$$\begin{aligned} &2(x_1 + x_2) + x_{10} + x_{11} \leq 10, \quad 2(x_1 + x_3) + x_{10} + x_{12} \leq 10, \quad 2(x_2 + x_3) + x_{11} + x_{12} \leq 10, \\ &-8x_1 + x_{10} \leq 0, \quad -8x_2 + x_{11} \leq 0, \quad -8x_3 + x_{12} \leq 0, \quad -2x_4 - x_5 + x_{10} \leq 0, \quad -2x_6 - x_7 + x_{11} \leq 0, \\ &-2x_8 - x_9 + x_{12} \leq 0, \\ &0 \leq x_i \leq 1, i = 1, \dots, 9, \quad 0 \leq x_i \leq 100, i = 10, 11, 12, \quad 0 \leq x_{13} \leq 1. \end{aligned}$$

$$G_2(x) = x_1 + x_2 + x_3,$$

subject to

$$\begin{aligned} &1 - 0.0025(x_4 + x_6) \geq 0, \quad 1 - 0.0025(x_5 + x_7 - x_4) \geq 0, \quad 1 - 0.01(x_8 - x_5) \geq 0, \\ &x_1x_6 - 833.33252x_4 - 100x_1 + 83333.333 \geq 0, \quad x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \geq 0, \\ &x_3x_8 - 1250000 - x_3x_5 + 2500x_5 \geq 0, \\ &100 \leq x_1 \leq 10000, \quad 1000 \leq x_i \leq 10000, i = 2, 3, \quad 10 \leq x_i \leq 1000, i = 4, \dots, 8. \end{aligned}$$

$$G_3(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7,$$

subject to

$$\begin{aligned} &127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0, \quad 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0, \\ &196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0, \quad -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0, \\ &-10 \leq x_i \leq 10, i = 1, \dots, 7. \end{aligned}$$

$$G_4(x) = e^{x_1x_2x_3x_4x_5},$$

subject to

$$\begin{aligned} &x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10, \quad x_2x_3 - 5x_4x_5 = 0, \quad x_1^3 + x_3^3 = -1, \\ &-2.3 \leq x_i \leq 2.3, i = 1, 2, \quad -3.2 \leq x_i \leq 3.2, i = 3, 4, 5. \end{aligned}$$

$$G_5(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2$$

$$+ 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

subject to

$$\begin{aligned} &105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \geq 0, \quad -10x_1 + 8x_2 + 17x_7 - 2x_8 \geq 0, \quad 8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0, \\ &-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \geq 0, \quad -5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0, \\ &-x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \geq 0, \quad -0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_8^2 + x_6 + 30 \geq 0, \\ &3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0, \\ &-10 \leq x_i \leq 10, i = 1, \dots, 10. \end{aligned}$$

α GA も全ての問題について標準設定を用いた。 α 制約法に関するパラメータについては、制約満足度の結合演算を min 演算、制約満足度のパラメータを $b_i = b_j = 10000$ とした。 遺伝的アルゴリズムに関するパラメータについては、個体集団サイズ $N = 70$, 最大世代数 $T = 5000$, 交叉率 $P_c = 0.3$, $\beta_c = 1.0$, 最大期待値 $\eta^+ = 2.0$, 境界突然変異率 $P_b = 0.3/n$, ガウス突然変異 $P_G = 0.3/n$, $\beta_G = 0.01$ とした。 ただし, n は遺伝子長 (決定変数の数) であり, 問題によらず一個体あたりの確率を一定にするために, 交叉率および各突然変異率を n に反比例させた。

等式制約を含む G_4 については, GENOCOP5.0 では式 (13) を使用したが, α GA では制約満足度に式 (3) を使用し, α レベルの制御を行った。

Michalewicz の研究では, 10 回の試行を行っているが, 10 回程度では偶然良い結果が連続することもあるので, より厳密な評価を行うために, 本論文では 100 回の試行を行った。

5.2 実験結果

問題 $G_1 \sim G_5$ に対する実験結果を表2に示す。ただし、Optimal は各問題の最適値，best, average, worst, σ はそれぞれ100回の試行における最良値，平均値，最悪値，標準偏差であり，CPU は実行時間である。

最良値については，全ての場合について α GAの方が，優れており， $G_1 \sim G_3, G_5$ については，実行可能最良解を得ている。 G_4 については，両手法とも実行可能最良解を得られなかった。GENOCOP5.0では制約条件を緩和したため，得られた解は制約条件を 10^{-3} 程度逸脱している。それに比して， α レベルの制御による α GAの結果では，最良値の場合で 10^{-12} ，最悪値の場合で 10^{-10} ，平均値の場合で 10^{-11} 程度の制約条件からの逸脱に止まっており，GENOCOP5.0に比べて十分に実行可能解に近似した解が得られている。さらに，平均値と最悪値についても，全ての問題について α GAの方が優れている。したがって， α GAの方がGENOCOP5.0に比べて，探索精度の高い最適化アルゴリズムであると考えられる。

また，標準偏差が全ての問題について α GAの方が小さいことから，安定した探索を行う最適化アルゴリズムであると考えられる。

さらに，実行時間についても全ての問題で α GAの方がGENOCOP5.0に比べて3倍以上速く，高速な最適化アルゴリズムであると考えられる。

表 2: 実験結果

f	Optimal	Item	best	average	worst	σ	CPU(s)
G_1	-15.000	α GA	-15.000	-15.000	-15.000	0.000	6.43
		GCOP	-15.000	-11.009	-3.160	4.481	24.58
G_2	7049.33	α GA	7049.61	7340.57	9748.20	445.761	5.97
		GCOP	7061.38	8060.56	12978.76	935.146	19.22
G_3	680.63	α GA	680.631	680.654	680.694	0.015	5.84
		GCOP	680.632	680.708	681.456	0.108	18.51
G_4	0.05395	α GA	0.05395	0.23597	1.00433	0.291	6.20
		GCOP	0.05387	0.98964	8.82806	1.141	27.92
G_5	24.306	α GA	24.319	24.446	24.893	0.093	7.05
		GCOP	24.522	29.057	46.369	4.470	27.38

表3に各問題の上下限制約領域に対する実行可能領域の割合 $\rho = |F \cap S|/|S|$ を示した[11]。これによると，実行可能領域が比較的広い問題である G_3 については GENOCOP5.0 もある程度の解を得ているが，厳しい問題である G_4, G_5 においては α GAの方が圧倒的に優れている。このように， α GAは実行可能領域が厳しい場合においても精度の高い結果を得ることのできる方法である。

表 3: 実行可能領域の占有率 $\rho = |F \cap S|/|S|$

f	G_1	G_2	G_3	G_4	G_5
$\rho(\%)$	0.0111	0.0010	0.5121	0.0000	0.0000

6 むすび

遺伝的アルゴリズムに α 制約法を適用した α GAを提案した。様々なタイプの5種類の制約付き最適化問題に α GAを適用し，全ての問題で最適値に近い解が得られることを実験により確認し， α GAが精度の高い安定したアルゴリズムであることを示した。さらに，制約付き最適化問題を遺伝的アルゴリズムを利

- [14] 高濱徹行, 阪井節子: 制約付き非線形最適化手法 α 制約法によるファジー制御ルールの最適化, 電子情報通信学会論文誌, Vol. J82-A, No. 5, pp. 658-668 (1999).
- [15] 高濱徹行, 阪井節子: α 制約Simplex法によるファジー制御ルールの学習, 電子情報通信学会論文誌, Vol. J83-D-I, No. 7, pp. 770-779 (2000).
- [16] 高濱徹行, 阪井節子: α 制約遺伝的アルゴリズムによる制約付き最適化, 第62回情報処理学会全国大会講演論文集, Vol. 2, pp. 111-112 (2001).
- [17] Powell, M.: An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives, *Computer J.*, Vol. 7, pp. 155-162 (1964).
- [18] Nelder, J. and Mead, R.: A Simplex Method for Function Minimization, *J. Computer*, Vol. 7, pp. 308-313 (1965).
- [19] Koziel, S. and Michalewicz, Z.: A Decoder-based Evolutionary Algorithm for Constrained Parameter Optimization Problems, *Proc. of the 5th Parallel Problem Solving from Nature, Lecture Notes in Computer Science*, Vol. 1498, Amsterdam, pp. 231-240 (1998).
- [20] Koziel, S. and Michalewicz, Z.: Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization, *Evolutionary Computation*, Vol. 7, No. 1, pp. 19-44 (1999).
- [21] Baker, J.: Adaptive selection methods for genetic algorithms, *Proc. of the First International Conference on Genetic Algorithms and Their Applications*, Hillsdale, NJ, Lawrence Erlbaum Associates, pp. 101-111 (1985).
- [22] Bäck, T. and Hoffmeister, F.: Extended selection mechanisms in genetic algorithms, *Proc. of the Fourth International Conference on Genetic Algorithms*, San Mateo, Morgan Kaufmann, pp. 92-99 (1991).